Todas as entradas e saída dos algoritmos são utilizados o STDIN e STDOUT de cada linguagem, abaixo tem algumas dicas de como utilizar cada STDIN e STDOUT de cada linguagem.

**JavaScript**

Em JavaScript as funções de STDIN e STDOUT respectivamente são **gets** e **console.log**, a função gets é implementada internamente para auxiliar a entrada dos dados.

**Exemplo:**

let line = gets(); // Lê a linha de entrada

console.log(line); // Imprime o dado

**Java**

Em Java existe várias formas de implementar o STDIN e STDOUT recomendamos utilizar **BufferedReader** para o STDIN e o **System.out.println** para o STDOUT.

**Exemplo:**

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

StringTokenizer st = new StringTokenizer(br.readLine()); // Lê a linha de entrada

int a = Integer.parseInt(st.nextToken());

System.out.println(a); // Imprime o dado

**Python**

Em Python existe várias formas de implementar o STDIN e STDOUT recomendamos utilizar **sys.stdin.readline** para o STDIN e o **print** para o STDOUT.

**Exemplo:**

import sys

a = int(sys.stdin.readline()) // Lê a linha de entrada

print(a); // Imprime o dado

**Desafio**

Vamos criar uma classe chamada UsuarioTelefone para representar um usuário de telefone. Você pode definir um método especial e depois aplicar conceitos de encapsulamento nos atributos dentro da classe. Lembre-se que, cada usuário terá um nome, um número de telefone e um plano associado, neste desafio, simulamos três planos, sendo: Plano Essencial Fibra, Plano Prata Fibra e Plano Premium Fibra.

**Entrada**

Nome do usuário, número de telefone e plano.

**Saída**

Mensagem indicando que o usuário foi criado com sucesso.

**Exemplos**

A tabela abaixo apresenta exemplos com alguns dados de entrada e suas respectivas saídas esperadas. Certifique-se de testar seu programa com esses exemplos e com outros casos possíveis.

| **Entrada** | **Saída** |
| --- | --- |
| Ana (11) 91111-1111 Plano Essencial Fibra | Usuário Ana criado com sucesso. |
| Sofia (22) 92222-2222 Plano Prata Fibra | Usuário Sofia criado com sucesso. |
| Pedro (33) 93333-3333 Plano Premium Fibra | Usuário Pedro criado com sucesso. |

# TODO: Crie uma classe UsuarioTelefone.

# TODO: Defina um método especial `\_\_init\_\_`, que é o construtor da classe.

# O método `\_\_init\_\_`, irá inicializar os atributos da classe: `nome`, `numero` e `plano`.

# TODO: Aplique o conceito de encapsulamento, onde os atributos serão encapsulados dentro da classe.

# A classe `UsuarioTelefone` define um método especial `\_\_str\_\_`, que retorna uma representação em string do objeto.

def \_\_str\_\_(self):

return f"Usuário {self.nome} criado com sucesso."

# Entrada:

nome = input()

numero = input()

plano = input()

# TODO: Crie um novo objeto `UsuarioTelefone` com os dados fornecidos:

print(usuario)